# ON THE EXPECTED COMPLEXITY OF INTEGER LEAST-SQUARES PROBLEMS

*Babak Hassibi*

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125

*Haris Vikalo*

Information Systems Laboratory
Stanford University
Stanford, CA 94305

**ABSTRACT**

The problem of finding the least-squares solution to a system of linear equations where the unknown vector is comprised of integers, but the matrix coefficient and given vector are comprised of real numbers, arises in many applications: communications, cryptography, GPS, to name a few. The problem is equivalent to finding the closest lattice point to a given point and is known to be NP-hard. In communications applications, however, the given vector is not arbitrary, but rather is an unknown lattice point that has been perturbed by an additive noise vector whose statistical properties are known. Therefore in this paper, rather than dwell on the worst-case complexity of the integer-least-squares problem, we study its expected complexity, averaged over the noise and over the lattice. For the "sphere decoding" algorithm of Fincke and Pohst we find a closed-form expression for the expected complexity and show that for a wide range of noise variances the expected complexity is polynomial, in fact often sub-cubic. Since many communications systems operate at noise levels for which the expected complexity turns out to be polynomial, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented in real-time—a result with many practical implications.

## 1. THE INTEGER LEAST-SQUARES PROBLEM

In this paper we shall be concerned with the following so-called *integer least-squares problem*

$$\min_{s\in\mathcal{Z}^m} \|x - Hs\|_2 , \qquad (1)$$

where $x \in \mathcal{R}^n$, $H \in \mathcal{R}^{n\times m}$, and $\mathcal{Z}^m$ denotes the $m$-dimensional integer lattice. Often, the search space is a (finite) subset of a lattice, $\mathcal{D} \subset \mathcal{Z}^m$, in which case we have

$$\min_{s\in\mathcal{D}\subset\mathcal{Z}^m} \|x - Hs\|_2 . \qquad (2)$$

Integer least-squares problems arise in many communications problems (see, e.g., [1, 2]), as well as in global positioning systems (GPS) [3]. Problems (1) and (2) are well known to be NP-hard, both in a worst-case and in an average sense [4]. In fact, there is a whole family of public-key cryptosystems based on the NP-hardness of the integer least-squares problem [5, 6].

### 1.1. Heuristic Methods

All practical systems resort to approximations and/or heuristics such as the following:

1. Solve the unconstrained least-squares problem and then round-off to the closest integer. This is called *zero-forcing equalization* or Babai estimation [7].
2. *Nulling and cancelling.* This is also known as *decision-feedback equalization.*
3. *Nulling and cancelling with optimal ordering:* Perform nulling/cancelling, ordered from the "strongest" to the "weakest" signal (see [8, 9]).

All the above heuristic solutions require $O(m^3)$ computations and are exact only if the columns of $H$ are orthogonal, which is rarely the case. Orthogonalizing the columns of $H$ via a QR decomposition, or otherwise, generally destroys the lattice structure. *Lattice reduction* methods such as the LLL (Lenstra, Lenstra and Lovasz) algorithm [7] can be used to "orthogonalize as much as possible" the matrix $H$, while preserving the lattice structure. While this may lead to some improvement in the solution of (1), it is not useful for (2) since it destroys the properties of the subset $\mathcal{D} \subset \mathcal{Z}^m$.

## 2. SPHERE DECODING

There also exist exact methods that are a bit more sophisticated than performing a full search over the entire integer lattice [10]. One is the sphere decoding algorithm of Fincke and Pohst [11], which has recently been suggested for various communications problems [1, 12]. The main idea in sphere decoding is to search over only lattice points that lie in a certain hypersphere of radius $r$ around $x$, thereby reducing the required computations. Clearly, the closest lattice

0-7803-7402-9/02/$17.00 ©2002 IEEE

point inside the hypersphere will also be the closest lattice point for the whole lattice. Two questions come up.

1. *How to choose r?* $r$ too large, we obtain too many points. $r$ too small, we obtain no points.
2. *How can we tell which lattice points are inside the sphere?* If this requires testing each lattice point, then there is no point in sphere decoding.

Sphere decoding does not really address the first question. However, it does propose an efficient way to answer the second one. Note that $s$ lies in a sphere of radius $r$ iff

$$r^2 \geq \|x - Hs\|^2 = (s-\hat{s})^* H^* H (s-\hat{s}) + \|x\|^2 - \|H\hat{s}\|^2 .$$

where $\hat{s} = H^\dagger x$. Introducing the $QR$ decomposition $H = QR$, and defining $r'^2 = r^2 - \|x\|^2 + \|H\hat{s}\|^2$,

$$\begin{aligned} r'^2 &\geq (s-\hat{s})^* R^* R (s-\hat{s}) = \sum_{i=1}^{m} r_{ii}^2 \left( s_i - \hat{s}_i + \sum_{j=i+1}^{m} \frac{r_{ij}}{r_{ii}} (s_j - \hat{s}_j) \right)^2 \\ &= r_{mm}^2 (s_m - \hat{s}_m)^2 + \\ &\quad r_{m-1,m-1}^2 \left( s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}} (s_m - \hat{s}_m) \right)^2 + \ldots \quad (3) \end{aligned}$$

A necessary condition for $s$ to lie inside the sphere is therefore that $r_{mm}^2 (s_m - \hat{s}_m)^2 \leq r'^2$. This condition is equivalent to $s_m$ belonging to the interval

$$\left\lceil \hat{s}_m - \frac{r'}{r_{mm}} \right\rceil \leq s_m \leq \left\lfloor \hat{s}_m + \frac{r'}{r_{mm}} \right\rfloor . \qquad (4)$$

Of course, (4) is by no means sufficient. For every $s_m$ satisfying (4), defining $r'^2_{m-1} = r'^2 - r_{mm}^2 (s_m - \hat{s}_m)^2$, a stronger necessary condition can be found by looking at the first two terms in (3), which leads to $s_{m-1}$ belonging to the interval

$$\left\lceil \hat{s}_{m-1|m} - \frac{r'_{m-1}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \hat{s}_{m-1|m} + \frac{r'_{m-1}}{r_{m-1,m-1}} \right\rfloor .$$

One can continue in a similar fashion for $s_{m-2}$, and so on, to obtain all points inside the sphere.

### The Sphere Decoding Algorithm

*Input:* $R, x, \hat{s}, r$.

1. Set $k = m$, $r'^2_m = r^2 - \|x\|^2 + \|H\hat{s}\|^2$, $\hat{s}_{m|m+1} = \hat{s}_m$
2. (Bounds for $s_k$) Set $z = \frac{r'_k}{r_{kk}}$, $UB(s_k) = \lfloor z + \hat{s}_{k|k+1} \rfloor$, $s_k = \lceil -z + \hat{s}_{k|k+1} \rceil - 1$
3. (Increase $s_k$) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$ go to 5, else to 4.
4. (Increase $k$) $k = k + 1$ and go to 3.
5. (Decrease $k$) If $k = 1$ go to 6. Else $k = k - 1$, $\hat{s}_{k|k-1} = \hat{s}_k + \sum_{j=k+1}^{m} \frac{r_{kj}}{r_{kk}} (s_j - \hat{s}_j)$, $r'^2_k = r'^2_{k+1} - r^2_{k+1,k+1} (s_{k+1} - \hat{s}_{k+1|k+2})^2$.
6. Solution found. Save $s_k$ and go to 3.

**Remark:** Rather than search over all lattice points in a sphere of radius $r$ and dimension $m$, the algorithm searches over all lattice points in spheres of radius $r$ and dimensions $1, 2, \ldots, m$. The algorithm therefore constructs a tree, where the branches in the $k$-th level of the tree correspond to the lattice points inside the sphere of radius $r$ and dimension $k$. This is depicted in Fig. 1.

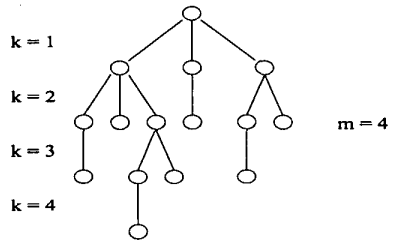


**Fig. 1.** Tree generated by sphere decoding algorithm.

### 2.1. A First Look at Complexity

The complexity of the sphere decoding algorithm depends on the size of the tree in Fig. 1, which is equal to the sum of the number of points in spheres of radius $r$ and dimensions $k = 1, \ldots, m$. For an arbitrary point $x$, the expected number of lattice points inside a $k$-dimensional sphere of radius $r$ is proportional to its volume, $\frac{\pi^{k/2}}{\Gamma(k/2+1)} r^k$. Therefore the expected total number of points is

$$\sum_{k=1}^{m} \frac{\pi^{k/2}}{\Gamma(k/2+1)} r^k > \sum_{k=1}^{\frac{m}{2}} \frac{\pi^k}{\Gamma(k+1)} r^{2k} \approx e^{\pi r^2}, \text{ for large } m.$$

To have a nonvanishing probability of finding a point in the $m$-dimensional sphere, its volume must be $\frac{\pi^{m/2}}{(m/2)!} r^m = O(1)$. But from Stirling's formula this implies that $r^2 = O(m)$ and that the expected complexity of the algorithm is exponential, $e^{O(m)}$.

## 3. A RANDOM MODEL

Although not unexpected, the above is a discouraging result. In communications applications, however, the vector $x$ is not arbitrary, but rather is a lattice point perturbed by additive noise with known statistical properties. Thus, we will assume $x = Hs + v$, where the entries of $v$ are independent $N(0, \sigma^2)$ random variables. We will also assume that the lattice-generating-matrix $H$ is random and is comprised of independent $N(0,1)$ entries. We further assume, for simplicity, that $m = n$. (The more general case of $m \neq n$ can also be studied without much more effort.)

The first by-product of these assumptions is a method to determine the desired radius $r$. Note that $\|v\|^2 = \|x - Hs\|^2$ is a $\Xi^2$ random variable with $m/2$ degrees of freedom. Thus we may choose the radius $r^2 = \alpha m \sigma^2$ in such a way that we find a lattice point with high probability:

$$\int_0^{\alpha m} \frac{\lambda^{m/2-1}}{\Gamma(m/2)} e^{-\lambda} d\lambda = 0.99, \quad \text{say.}$$

The expected complexity can now be given by

$$\sum_{k=1}^{m} \underbrace{(\text{expected \# of points in } k\text{-sphere of radius } r)}_{\triangleq E_p(k, r^2 = \alpha m \sigma^2)} \cdot \underbrace{(\text{flops/point})}_{2k+17}$$

We need to compute $E_p(k, r^2)$. If the lattice point $s_t$ was transmitted and the vector $x = Hs_t + v$ received, the probability that the lattice point $s_a$ lies in a hypersphere of radius $r$ around $x$ is

$$\gamma\left(\frac{r^2}{\sigma^2 + \|s_a - s_t\|^2}, \frac{k}{2}\right) = \int_0^{\frac{r^2}{\sigma^2 + \|s_a - s_t\|^2}} \frac{\lambda^{k/2-1}}{\Gamma(k/2)} e^{-\lambda} d\lambda.$$

The above probability depends only on $\|s_a - s_t\|^2 = \|s\|^2$, i.e., on the squared norm of an arbitrary lattice point in the $k$-dimensional lattice. It is thus straightforward to see that

$$E_p(k, r^2) = \sum_{n=0}^{\infty} \gamma\left(\frac{r^2}{\sigma^2 + n}, \frac{k}{2}\right) \cdot (\text{\# of lattice points with } \|s\|^2 = n).$$

Since $\|s\|^2 = s_1^2 + \ldots + s_k^2$, we basically, need to figure out how many ways a non-negative integer $n$ can be represented as the sum of $k$ squared integers. This is a classic problem in number theory and the solution is denoted by $r_k(n)$ [13]. There exist a plethora of results on how to compute $r_k(n)$. We only mention one here: $r_k(n)$ is given by the coefficient of $x^n$ in the expansion

$$\left(1 + 2\sum_{m=1}^{\infty} x^{m^2}\right)^k = 1 + \sum_{n=1}^{\infty} r_k(n) x^n.$$

The above arguments lead to the following result.

**Theorem 1 (Expected complexity for problem (1))** *Under the aforementioned assumptions, the expected complexity of the sphere decoder for problem (1) is given by*

$$C(m, \sigma^2) = \sum_{k=1}^{m} (2k + 17) \sum_{n=0}^{\infty} r_k(n) \gamma\left(\frac{\alpha m \sigma^2}{\sigma^2 + n}, \frac{k}{2}\right),$$

*where $\alpha$ is such that $\gamma(\alpha m, m) = 1 - \epsilon$.*

It is often useful to look at the *complexity exponent*, $\frac{\log C(m, \sigma^2)}{\log m}$, which approaches a constant if the expected complexity is polynomial, and grows like $\frac{m}{\log m}$ if it is exponential. The complexity exponent is plotted as a function of $m$ for different values of the $\sigma^2$ in Fig. 2. As can be seen, for small enough noise the expected complexity is polynomial, whereas for large noise it is exponential.

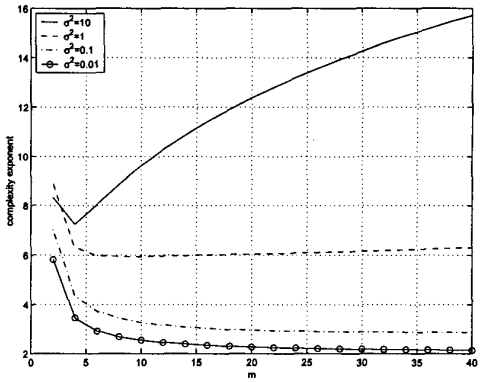


**Fig. 2.** *The complexity exponent as a function of $m$ for $\sigma^2 = 0.01, 0.1, 1, 10$.*

In communications problems, we are usually concerned with $L$-PAM constellations

$$\mathcal{D}_L^m = \left\{-\frac{L-1}{2}, -\frac{L-3}{2}, \ldots, \frac{L-3}{2}, \frac{L-1}{2}\right\}^m.$$

In this case, rather than the noise variance $\sigma^2$, one is interested in the SNR, $\rho = \frac{m(L^2-1)}{12\sigma^2}$. For such constellations, we have the following result.

**Theorem 2 (Expected complexity for problem (2))** *Under the aforementioned assumptions, the expected complexity of the sphere decoder for problem (2) for a 2-PAM constellation is*

$$C(m, \rho) = \sum_{k=1}^{m} (2k+17) \sum_{n=0}^{k} \binom{k}{n} \gamma\left(\frac{\alpha m}{1 + \frac{12\rho n}{m(L^2-12)}}, \frac{k}{2}\right).$$

*For a 4-PAM constellation it is*

$$\sum_{k=1}^{m} (2k+17) \sum_{n} \frac{1}{2^k} \sum_{l=0}^{k} \binom{k}{l} g_{kl}(n) \gamma\left(\frac{\alpha m}{1 + \frac{12\rho n}{m(L^2-12)}}, \frac{k}{2}\right),$$

*where $g_{kl}(n)$ is the coefficient of $x^n$ in the polynomial $(1 + x + x^4 + x^9)^l (1 + 2x + x^4)^{k-l}$. Similar expressions can be obtained for 8-PAM, 16-PAM, etc., constellations.*

Fig. 3 shows the complexity exponent as a function of $m$ for $\rho = 20$db, for different $L$-PAM constellations with $L = 2, 4, 8, 16$. For low rates (i.e., small constellations) the expected complexity is polynomial, whereas for high rates (i.e., large constellations) it is exponential. Simulation results suggest that the complexity is polynomial as long as the rate is sufficiently, but not necessarily all that much, below the Shannon capacity corresponding to the SNR. Since this is the regime at which most communication systems operate, it suggests that ML decoding can be feasible.

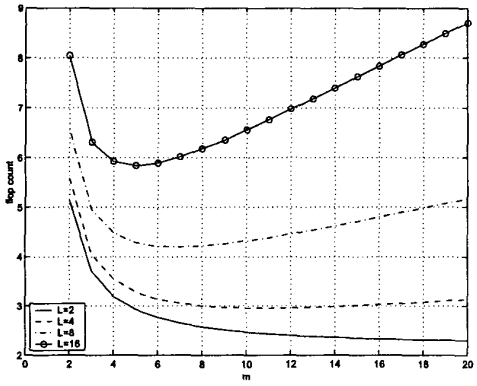


**Fig. 3**. *The complexity exponent as a function of* $m$ *for* $\rho = 20db$ *and* $L = 2, 4, 8, 16$.

Fig. 4 shows the improvement in performance of sphere decoding over nulling and cancelling for a certain multi-antenna space-time code corresponding to $m = 64$ (for the details see [2]). The complexity of ML decoding via the sphere decoder here is comparable to nulling and cancelling, whereas the performance improvement is significant. In fact, here the lattice subset has $10^{38}$ points and the sphere decoder yields the ML estimate in (roughly) cubic time.

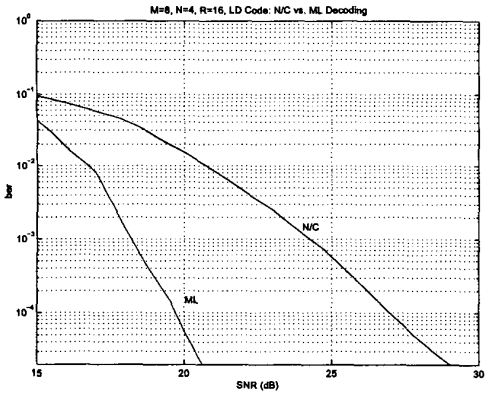


**Fig. 4**. *Sphere decoder (ML) vs. nulling and cancelling with optimal ordering (NC).*

## 4. CONCLUSION

In many communication problems, maximum-likelihood detection reduces to solving an integer least-squares problem. In such applications ML detection is rarely performed, on the grounds that it requires exponential complexity and is therefore computationally intractable. In this paper we obtained a closed-form expression for the expected complexity of sphere decoding in terms of the noise variance, the dimension of the lattice, and (for subsets of lattices) the constellation. It turns out that over a wide range of noise variances and dimensions the expected complexity is often cubic or sub-cubic. Since many communications systems operate at noise levels for which this is the case, this suggests that maximum-likelihood decoding, which was hitherto thought to be computationally intractable, can in fact be implemented with complexity similar to heuristic methods, but with siginificant performance gains—a result with many practical implications.

## 5. REFERENCES

[1] C. Brutel and J. Boutros, "Euclidean space lattice decoding for joint detection in CDMA systems," in *Proc. of the 1999 IEEE Info. Thy. and Comm. Workshop*, p. 129, 1999.

[2] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *submitted to IEEE Trans. Info. Theory*, 2000. Download available at `http://mars.bell-labs.com`.

[3] A. Hassibi and S. Boyd, "Integer parameter estimation in linear models with applications to GPS," *IEEE Transactions on Signal Processing*, vol. 46, pp. 2938–52, November 1998.

[4] M. Ajtai, "The shortest vector problem in $L_2$ is NP-hard for randomized reductions," in *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp. 10–19, 1998.

[5] M. Ajtai, "Generating hard instances of lattice problems," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pp. 99–108, 1996.

[6] O. Goldreich, S. Goldwasser, and S. Halevi, "Public-key cryptosystems from lattice reduction problems," in *Advances in Cryptology - CRYPTO97. 17th Ann. Int. Crypto. Conf.*, pp. 112–31, 1997.

[7] M. Grotschel, L. Lovasz, and A. Schrijver, *Geometrical Algorithms and Combinatorial Optimization*. New York, NY: Springer-Verlag, 1993.

[8] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs. Tech. J.*, vol. 1, no. 2, pp. 41–59, 1996.

[9] B. Hassibi, "An efficient square-root algorithm for BLAST," *submitted to IEEE Trans. Sig. Proc.*, 1999. Download available at `http://mars.bell-labs.com`.

[10] R. Kannan, "Improved algorithms on integer programming and related lattice problems", in *Proc. 15th Annu. ACM Symp. on Theory of Computing*, 1983, pp. 193-206.

[11] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, pp. 463–471, April 1985.

[12] M. O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Comm. Let.*, pp. 161–163, May 2000.

[13] G. Hardy, *Ramanujan: Twelve Lectures*. Chelsea Publishing, 1940.